WHAT IS CLAIMED IS:

1. A middleware layer configured to facilitate communication between a speech-related application and a speech-related engine, comprising:

a speech component having an application-independent interface configured to be coupled to the application and an engine-independent interface configured to be coupled to the engine and at least one processing component configured to perform speech related services for the application and the engine.

2. The middleware layer of claim 1 wherein the speech component includes a plurality of processing components associated with a plurality of different processes, and wherein the speech component further comprises:

a marshaling component, configured to access at least one processing component in each process and to marshal information transfer among the processes.

3. The middleware layer of claim 1 wherein the speech component has an interface configured to be coupled to an audio device, and wherein the speech component further comprises:

09751836.122900

a format negotiation component configured to negotiate a data format of data used by the audio device and data used by the engine.

4. The middleware layer of claim 3 wherein the format negotiation component is configured to reconfigure the audio device to change the data format of the data used by the audio device.

5. The middleware layer of claim 3 wherein the format negotiation component is configured to reconfigure the engine to change the data format of the data used by the engine.

6. The middleware layer of claim 3 wherein the format negotiation component is configured to invoke a format converter to convert the data format of data between the engine and the audio device to a desired format based on the data format used by the audio device and the data format used by the engine.

7. The middleware layer of claim 1 wherein the processing component comprises:

a lexicon container object configured to contain a plurality of lexicons and to provide a lexicon interface to the engine to represent the plurality of lexicons as a single lexicon to the engine.

8. The middleware layer of claim 7 wherein the lexicon container object is configured to, once instantiated, load one or more user lexicons and one or more application lexicons from a lexicon data store.

9. The middleware layer of claim 8 wherein the lexicon interface is configured to be invoked by the engine to add a lexicon provided by the engine.

10. The middleware layer of claim 1 wherein the processing component comprises:

a site object having an interface configured to receive result information from the engine.

11. The middleware layer of claim 1 wherein the engine comprises a text-to-speech (TTS) engine and wherein the processing component comprises:

a first object having an application interface and an engine interface.

12. The middleware layer of claim 11 wherein the application interface exposes a method configured to receive engine attributes from the application and instantiate a specific engine based on the engine attributes received.

13. The middleware layer of claim 11 wherein the application interface exposes a method configured to

receive audio device attributes from the application and instantiate a specific audio device based on the audio device attributes received.

14. The middleware layer of claim 11 wherein the first object includes a parser configured to receive input data to be synthesized and parse the input data into text fragments.

15. The middleware layer of claim 11 wherein the engine interface is configured to call a method exposed by the engine to begin synthesis.

16. The middleware layer of claim 1 wherein the engine comprises a speech recognition (SR) engine and wherein the processing component comprises:

a first object having an application interface and an engine interface.

17. The middleware layer of claim 16 wherein the application interface exposes a method configured to receive recognition attributes from the application and instantiate a specific speech recognition engine based on the engine attributes received.

18. The middleware layer of claim 16 wherein the application interface exposes a method configured to receive audio device attributes from the application

09751836.122900

and instantiate a specific audio device based on the audio device attributes received.

19. The middleware layer of claim 16 wherein the application interface exposes a method configured to receive an alternate request from the application and to configure the speech component to retain alternates provided by the SR engine for transmission to the application based on the alternate request.

20. The middleware layer of claim 16 wherein the application interface exposes a method configured to receive an audio information request from the application and to configure the speech component to retain audio information recognized by the SR engine based on the audio information request.

21. The middleware layer of claim 16 wherein the application interface exposes a method configured to receive bookmark information from the application identifying a position in an input data stream being recognized and to notify the application when the SR engine reaches the identified position.

22. The middleware of claim 16 wherein the engine interface is configured to call the SR engine to set acoustic profile information in the SR engine.

09754835.122900

23. The middleware of claim 16 wherein the engine interface is configured to call the SR engine to load a grammar in the SR engine.

24. The middleware of claim 16 wherein the engine interface is configured to call the SR engine to load a language model in the SR engine.

25. The middleware layer of claim 16 wherein the application interface exposes a method configured to receive a grammar request from the application and to instantiate a grammar object based on the grammar request.

26. The middleware layer of claim 25 wherein the grammar object includes a word sequence data buffer and an interface configured to provide the SR engine with access to the word sequence data buffer.

27. The middleware layer of claim 25 wherein the grammar object includes a grammar to be used by the SR engine.

28. The middleware layer of claim 27 wherein the grammar includes words, rules and transitions and wherein the grammar object includes an application interface and an engine interface.

09751836.122600

29. The middleware layer of claim 28 wherein the application interface exposes a grammar configuration method configured to receive grammar configuration information from the application and configure the grammar based on the grammar configuration information.

30. The middleware layer of claim 29 wherein the grammar configuration method is configured to receive rule activation information and activate or deactivate rules in the grammar based on the rule activation information.

31. The middleware layer of claim 29 wherein the grammar configuration method is configured to receive grammar activation information and enable or disable grammars in the grammar object based on the grammar activation information.

32. The middleware layer of claim 29 wherein the grammar configuration method is configured to receive word change data, rule change data and transition change data and change words, rules and transitions in the grammar in the grammar object based on the grammar received data.

33. The middleware layer of claim 28 wherein the engine interface is configured to call the SR engine to load the grammar in the SR engine.

09754835.122600

34. The middleware layer of claim 33 wherein the engine interface is configured to call the SR engine to update a configuration of the grammar in the SR engine.

35. The middleware layer of claim 33 wherein the engine interface is configured to call the SR engine to update an activation state of the grammar in the SR engine.

36. The middleware layer of claim 1 wherein the processing component further comprises:

a site object exposing an engine interface configured to receive information from the SR engine.

37. The middleware layer of claim 36 wherein the engine interface on the site object is configured to receive result information from the SR engine indicative of recognized speech.

38. The middleware layer of claim 36 wherein the engine interface on the site object is configured to receive update information from the SR engine indicative of a current position of the SR engine in an audio input stream to be recognized.

39. The middleware layer of claim 36 wherein the processing component further comprises:

a result object configured to obtain the result information from the site object and expose an interface configured to pass the result information to the application.

40. A multi-process speech recognition middleware layer configured to facilitate communication between a speech recognition (SR) engine and one or more applications, the middleware layer comprising:

a first process including:

a first context object having an application interface to enable application control of a first plurality of attributes of the speech recognition and an engine interface; and

a first grammar object having an application interface and an engine interface and storing a first grammar used by the first process; and

a second process including:

a second context object having an application interface to enable application control of a first plurality of attributes of the speech recognition and an engine interface; and

a second grammar object having an application interface and an engine interface and

09751836.122900

storing a second grammar used by the second process; and

a server process configured to receive result information provided by the SR engine and provide the result information to the first or second process, to which the result information belongs.

41. The multi-process speech recognition middleware layer of claim 40 wherein the first and second grammars each include a plurality of rules and further comprising:

a grammar engine configured to store a grammar indication indicating the grammar to which each of the plurality of rules belong.

42. The multi-process speech recognition middleware layer of claim 41 wherein the SR engine returns, along with the result information, a rule identifier identifying a rule which spawned the result information.

43. The multi-process speech recognition middleware layer of claim 44 wherein the grammar engine examines the rule identifier to determine a particular grammar to which the identified rule belongs.

44. The multi-process speech recognition middleware claim 43 layer of wherein server process queries the

first or second grammar object containing the particular grammar to receive an identity of an associated context object.

45. The multi-process speech recognition middleware claim 44 wherein the server process notifies the associated context object that the result belongs to its associated process.

46. The multi-process speech recognition middleware claim 40 wherein the SR engine returns preliminary information to the server process and wherein the server process is configured to notify the first and second context objects of the preliminary information.

47. A multi-voice speech synthesis middleware layer configured to facilitate communication between one or more applications and a plurality of text-to-speech (TTS) engines, comprising:

at least a first voice object having an application interface configured to receive TTS engine attribute information from the application and to instantiate first and second TTS engines based on the TTS attribute information, to receive a speak request requesting at least one of the TTS engines to speak a message, and to receive priority information associated with each speak request indicative of a precedence each speak request is to take, and wherein the first voice object has an engine interface configured to call a specified one of the first and second TTS engines to synthesize input data.

48. The multi-voice speech synthesis middleware layer of claim 47 wherein the at least first voice object is configured to receive a normal priority associated with a message and to call the TTS engines so the message with normal priority is spoken in turn.

49. The multi-voice speech synthesis middleware layer of claim 48 wherein the at least first voice object is configured to receive a speakover priority associated with a message and to call the TTS engines so the message with speakover priority is spoken at a same time as other currently speaking messages.

50. The multi-voice speech synthesis middleware layer of claim 49 wherein the at least first voice object is configured to receive an alert priority associated with a message and to call the TTS engines so the message with alert priority is spoken with precedence over messages with normal and speakover priority.

51. A method of updating a grammar configuration of a grammar used by a speech recognition (SR) engine based on update information from an application, comprising:

calling a first object in an application-independent, engine-independent middleware layer, between the SR engine and the application, with a pause request;

delaying return from the first object on a subsequent call from the SR engine;

receiving the update information from the application at the middleware layer;

passing the update information from the middleware layer to the SR engine; and

returning on the subsequent call from the SR engine.

09751836.122900

52. The method of claim 51 wherein receiving the update information comprises:

receiving word change data, rule change data and transition change data from the application; and

changing words, rules and transitions in a grammar in the middleware layer based on the word change data, rule change data and transition change data received.

53. A method of formatting data for use by a speech engine and an audio device, comprising

obtaining, at a middleware layer which facilitates communication between the speech engine and an application, a data format for data used by the engine;

obtaining, at the middleware layer, a data format of data used by the audio device;

determining, at the middleware layer, whether the engine data format and the audio data format are consistent; and

if not, utilizing the middleware layer to attempt to change the data format of the data used by at least one of the engine and the audio device.

54. The method of claim 53 and further comprising:

if the attempt to change the data format used by the at least one of the engine and the audio device is unsuccessful, invoking a format converter to change data format for data between the engine and the audio device to ensure the data formats are consistent.